REMARKS/ARGUMENTS

Amendments were made to the specification to update information regarding related applications identified therein and to correct an error noted by the Examiner. No new matter has been added by any of the amendments to the specification.

Claims 1-25 are pending in the present application. Claims 1-4, 6, 8-11, 13, 15-18, 20 and 22-25 were amended. No claims were added and no claims were canceled. Reconsideration of the claims is respectfully requested in view of the above amendments and the following comments.

I. 35 U.S.C. § 101

The Examiner has rejected claims 1-2, 4-9, 11-16, and 18-25 under 35 U.S.C. § 101 as being directed towards non-statutory subject matter. Specifically, the Examiner contends that the claims do not satisfy the requirements of 35 U.S.C. § 101 because they do not produce a practical, concrete, and tangible result.

By the present Amendment, claim 1 has been amended to positively recite "generating execution information relating to the instruction if the instruction is within the contiguous range of instructions". Similar amendments were made to independent claims 8, 15, 22, 24 and 25; and several of the dependent claims were also amended to more clearly recite a tangible result. Each of the claims in the application now clearly produces a practical, concrete and tangible result and fully satisfies the requirements of 35 U.S.C. § 101 in this regard.

Therefore, the rejection of the claims under 35 U.S.C. § 101 for failing to produce a practical, concrete, and tangible result has been overcome.

The Examiner has further rejected claims 24-25 under 35 U.S.C. § 101 as being directed towards non-statutory subject matter. Specifically, the Examiner asserts that claims 24 and 25 additionally fail to satisfy the requirements of 35 U.S.C. § 101 because the specification defines a computer readable medium as including "transmission-type media such as digital and analog communication links" which the Examiner contends is non-statutory. This rejection is respectfully traversed.

Applicants submit that no basis is present for holding a computer usable medium claim non-statutory because the medium may be allegedly "intangible." The MPEP states:

> In this context, "functional descriptive material" consists of **data structures** and computer programs **which impart functionality when employed as a computer component**. (The definition of "data structure" is "a physical or logical relationship among data elements, designed to support specific data manipulation functions." The New IEEE Standard Dictionary of Electrical and Electronics Terms 308 (5th ed. 1993).) "Nonfunctional descriptive material" includes but is not limited to music, literary works and a compilation or mere arrangement of data.

> **When functional descriptive material is recorded on some computer-readable medium it becomes structurally and functionally interrelated to the medium and will be statutory in most cases since use of technology permits the function of the descriptive material to be realized.** Compare *In re Lowry*, 32 F.3d 1579, 1583-84, 32 USPQ2d 1031, 1035 (Fed. Cir. 1994) (claim to data structure stored on a computer readable medium that increases computer efficiency held statutory) and *Warmerdam*, 33 F.3d at 1360-61, 31 USPQ2d at 1759 (claim to computer having a specific data structure stored in memory held statutory product-by-process claim) with *Warmerdam*, 33 F.3d at 1361, 31 USPQ2d at 1760 (claim to a data structure *per se* held nonstatutory). **(emphasis added)**

MPEP 2106 (IV)(B)(1)

Claims 24-25 recite clearly functional descriptive material since they impart functionality when employed as a computer component. Moreover, the functional descriptive material of claims 24-25 is recorded on "some" computer-readable medium.

In the above context, the term "some" means "any" computer-readable medium. The MPEP does not draw any distinctions between one type of media that is considered to be statutory and another type of media that is considered to be non-statutory. To the contrary, the MPEP clearly states that as long as the functional descriptive material is in "some" computer-readable medium, it should be considered statutory. The only exceptions to this statement in the MPEP are functional descriptive material that does not generate a useful, concrete and tangible result, e.g., functional descriptive material composed completely of pure mathematical concepts that provide no practical result. Claims 24-25 as amended herein clearly recite a useful, concrete and tangible result as indicated above, and do not just recite some disembodied mathematical concept or abstract idea.

Thus, claims 24-25 are directed to functional descriptive material that provides a useful, concrete and tangible result, and which is embodied on "some" computer-readable medium. Therefore, claims 24-25 are statutory and the further rejection of those claims under 35 U.S.C. § 101 has also been overcome.

II. 35 U.S.C. § 102, Anticipation

The Examiner has rejected claims 1-25 under 35 U.S.C. § 102(b) as being anticipated by Merten et al., "A Hardware-Driven Profiling Scheme for Identifying Program Hot Spot to Support Runtime Optimization." This rejection is respectfully traversed.

In rejecting the claims, the Examiner states:

> As per Claim 1: Merten discloses,
> *A method in a data processing system for monitoring execution of instructions, the method comprising:*
> *identifying an instruction for execution;*

determining whether the instruction is within a contiguous range of instructions (In run-time of a program, Merten depicts a code block: See left col., p. 137:2-18; depicts a code region: See left col., sec. 2, p. 138: These read limitation, "*contiguous range of instructions*". For execution, when the execution falls in the determined code region, it is known that every executed instruction in the region is within, and this is the purpose in which Merten runs the code for monitoring and identifying hotspots)*;*
and identifying execution information relating to the instruction if the instruction is within the contiguous range of instructions (Merten uses profiling to identify *execution information*, and to particularly identify the executions of branches within that code region: See sec. 2.1.1, p. 139).

Office Action dated June 5, 2006, pages 4-5.

Claim 1 of the present application, as amended herein is as follows:

1. A method in a data processing system for monitoring execution of instructions, the method comprising:
identifying an instruction for execution;
determining whether the instruction is within a contiguous range of instructions; and
generating execution information relating to the instruction if the instruction is within the contiguous range of instructions.

A prior art reference anticipates a claimed invention under 35 U.S.C. § 102 only if every element of the claimed invention is identically shown in that single prior art reference, arranged as they are in the claims. *In re Bond*, 910 F.2d 831, 832, 15 U.S.P.Q.2d 1566, 1567 (Fed. Cir. 1990). All limitations of a claimed invention must be considered when determining patentability. *In re Lowry*, 32 F.3d 1579, 1582, 32 U.S.P.Q.2d 1031, 1034 (Fed. Cir. 1994). Anticipation focuses on whether a claim reads on the product or process a prior art reference discloses, not on what the reference broadly teaches. *Kalman v. Kimberly-Clark Corp.,* 713 F.2d 760, 218 U.S.P.Q. 781 (Fed. Cir. 1983).

Applicants respectfully submit that Merten does not identically show every element of the claimed invention arranged as they are in the claims; and, accordingly, does not anticipate the claims. With respect to claim 1, in particular, Applicants respectfully submit that Merten does not teach or suggest any of the claimed steps of "identifying an instruction for execution", "determining whether the instruction is within a contiguous range of instructions", or "generating execution information relating to the instruction if the instruction is within the contiguous range of instructions".

Merten is directed to a mechanism for identifying program "hot spots" to support runtime optimization. In rejecting the claims as being anticipated by Merten, the Examiner refers particularly to left column, page 137, lines 2-18 and left column, section 2, page 138 of Merten, reproduced below for the convenience of the Examiner:

We have observed that many applications exhibit behavior conducive to runtime profiling and optimization. For example, program execution often occurs in distinct phases, where each phase consists of a set of code blocks that are executed with a high degree of temporal locality. When a collection of intensively executed blocks also has a small static footprint, it represents a highly favorable opportunity for runtime optimization. We will refer to such sets of blocks and their corresponding periods of execution as *hot spots*. A runtime optimizer can take advantage of execution phases by isolating a group of hot spots that are active for each phase. Ideally, aggressive optimized code would be deployed early in the phase and the optimized code used until execution shifts to another phase. Optimized hot spots that are no longer active may then be discarded, if necessary, to reclaim memory space for newly optimized code.

Our proposed hot spot detection scheme uses three criteria to classify a region of code as a hot spot. First, the region must have a small static code size to facilitate rapid optimization. Second, the hot spot must be active over a certain minimum time interval so that it is likely to have an opportunity to benefit from runtime optimization. Finally, the instructions in the selected region of code must account for a large majority of the dynamic execution during its active time interval. These three criteria are sufficient to detect code regions that can benefit most from runtime optimization without placing unnecessary restrictions on the type of hot spots that can be identified.

Nowhere in the above recitations or elsewhere in Merten is it described that an instruction is identified for execution, and that it is determined whether the identified instruction is within a contiguous range of instructions. Instead, Merten describes sets of code blocks that are executed with a high degree of temporal locality ("hot spots"), and describes three criteria by which a region of code is specified as a hot spot "that can benefit most from runtime optimization". As indicated above, the criteria include having a small static code size, being active over some time interval, and accounting for a large majority of dynamic execution during its active time interval. Merten nowhere discloses "identifying an instruction for execution" (nor, in fact, has the Examiner referred to any portion of Merten as disclosing this feature), and also does not disclose or suggest "determining whether the instruction is within a contiguous range of instructions". (Emphasis added.) Merten does not make a determination whether an instruction identified for execution is within a contiguous range of instructions, but, instead, Merten selects a set of code blocks based on the three criteria described above.

For similar reasons, Merten also does not disclose or suggest "generating execution information relating to the instruction if the instruction is within the contiguous range of instructions" as recited in claim 1. Again, Merten selects a set of code blocks based on specified criteria, but does not generate execution information relating to an instruction that is identified for execution, and that has been determined to be within a contiguous range of instructions. Merten, instead, identifies a set of code blocks "that can benefit most from runtime optimization" based on the criteria specified in Merten.

Claim 1, accordingly, is not anticipated by Merten and patentably distinguishes over Merten in its present form.

Independent claims 15 and 24 recite similar subject matter as claim 1, and are also not anticipated by Merten for similar reasons as discussed above with respect to claim 1.

Claims 2-7 depend from and further restrict claim 1, and claims 16-21 depend from and further restrict claim 15. These claims are also not anticipated by Merten at least by virtue of their dependency. In addition, many of these claims recite additional subject matter that is not disclosed by Merten. For example, claim 4 depends from claim 1 and further recites the steps of "determining whether the instruction is within a second contiguous range of instructions"; and "generating the execution information relating to the instruction if the instruction is within the second contiguous range of instructions". The Examiner refers to statements in Section 2.1.1 of Merten relating to monitoring short intervals of a branch/candidate branch, as disclosing these steps. Applicants respectfully disagree. Again, Merten does not disclose that a determination is made whether "the instruction [that was identified for execution and that was determined to be within a contiguous range of instructions] is within a second contiguous range of instructions", and then "generating the execution information relating to the instruction if the instruction is within the second contiguous range of instructions". Merten, at best, discloses profiling frequently executed branches that "may be part of a potential hot spot", and this is quite different than "determining whether the instruction is within a second contiguous range of instructions; and generating the execution information relating to the instruction if the instruction is within the second contiguous range of instructions" as recited in claim 4. Claim 4, accordingly, is not anticipated by Merten in its own right as well as by virtue of its dependency.

Independent claim 8, as amended herein is as follows:

> 8. A method in a data processing system for monitoring access to data in memory locations, the method comprising:
> identifying an access to data in a memory location;
> determining whether the memory location is within a contiguous range of memory locations; and
> generating information relating to the memory location if the memory location is within the contiguous range of memory locations.

Merten nowhere discloses or suggests "identifying an access to data in a memory location" and "determining whether the memory location is within a contiguous range of memory locations" as recited in claim 8, nor has the Examiner identified any such disclosure in Merten. Similarly, Merten does not disclose "generating information <u>relating to the memory location</u> if the memory location is within the contiguous range of memory locations" as also recited in claim 8. The Examiner states only that claim 8 is rejected on the same rationale as claim 1. The recitations in Merten referred to by the Examiner, however, are not related to monitoring memory location accesses, and claim 8 patentably distinguishes over Merten in its present form.

Independent claims 22 and 25 recite similar subject matter as claim 8, and are also not anticipated by Merten for similar reasons as discussed above with respect to claim 8. Claims 9-14 depend from and further restrict claim 8, and claim 23 depends from and further restricts claim 22. These claims are also not anticipated by Merten at least by virtue of their dependency.

Therefore, the rejection of claims 1-25 under 35 U.S.C. § 102 has been overcome.

III. <u>Conclusion</u>

For all the above reasons, it is respectfully urged that claims 1-25 are allowable in their present form, and that this application is now in condition for allowance. It is, accordingly, respectfully requested that the Examiner so find and issue a Notice of Allowance in due course.

The Examiner is invited to call the undersigned at the below-listed telephone number if in the opinion of the Examiner such a telephone conference would expedite or aid the prosecution and examination of this application.

DATE: September 5, 2006

Respectfully submitted,

<u>/Gerald H. Glanzman/</u>
Gerald H. Glanzman
Reg. No. 25,035
Yee & Associates, P.C.
P.O. Box 802333
Dallas, TX 75380
(972) 385-8777
Attorney for Applicants